بسم الله الرحمن الرحيم

((سبحانك لا علم لنا إلا ما علمتنا إنك أنت العليم الحكيم))

أسهل الطرق للبدء مع ++C

إعداد وتأليف : المبرمج العرماني

بكالوريوس علوم حاسب (Computer Science)

المملكة العربية السعودية – المنطقة الغربية

alarmani_programmer@hotmail.com

تم الإنتهاء من إعداد وتأليف هذا الكتاب في تاريخ :

٢٠٠٧/٦/٤ م – ١٤٢٨/٥/١٨ هـ

ملاحظـــــــــات حول هذا العمـــــــل :

١- إنطلاقاً من مبدأ لا خير في كاتم العلم شرعت في تأليف هذا الكتاب حول لغة السي ++ سائلاً الله الأجر والمثوبة فهو عمل خالص لوجه الله .
٢- طريقة الشرح في هذا الكتاب هي طريقة تطبيقية بحتة بحيث نبدأ بالكود ثم شرح كل كبيرة وصغيرة موجودة فيه وهذه الطريقة وجدتها أسرع في الفهم وأسهل في التعلم وتضيف فهم عميق للغة مقارنة بطريقة الشرح التقليدية .
٣- يمكن لكل قارئ لهذا الكتاب أن يقوم بنسخه أو توزيعه سواءا على صورته هذه أو غيرها ولكن أتمنى أن يتم ذكر الكاتب أو المؤلف عرفاناً لما قام به من جهد حتى وصوله للجميع وإنطلاقاً من مبدأ نسبة الشيء لصاحبه .
٤- قمت بإعداد وتأليف هذا الكتاب إعتماداً على مصادر مطبوعة علمية باللغتين العربية والانجليزية وليس نقلاً ولا نسخاً بحيث أعتمدت على طريقة شرحي الخاصة من خلال فهمي للغة إذن فهو مجهود شخصي مني بحت .
٥- هذا العمل موجه إلى فئة المبتدئين مع السي ++ .
٦- العمل إن شاء الله سوف يكون على الإصدار السادس وهو Microsoft Visual C++ 6.0 .
٧- هذا العمل يعتبر أرضية خصبة لإحتراف السي++ .
٨- أخيراً : أتمنى أن لاتنسوني من خالص دعائكم بالصحة والعافية والتوفيق لي ولوالدي ولجميع المسلميـــن .



١- مقدمة عامة عن لغة السي ++ :

تعتبر لغة السي++ من اللغات عالية المستوى high level languages وهذه اللغات تتصف بالمرونة وقابلية نقلها على أي آلة والسهولة .

والسي ++ هي لغة برمجة عامة الأغراض لذا فهي تستخدم في الكثير من التطبيقات كما أن طريقة الكتابة بها تتسم بالسهولة .

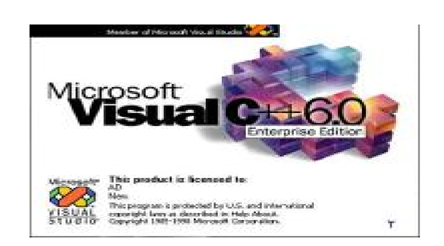
فعند كتابة البرنامج بلغة السي++ نقوم بترجمته عن طريق compile وهذا برنامج يقوم بتحويل الكود المصدري إلى كود بلغة الآلة . ومن ثم نقوم بتنفيذه عن طريق execute وهو الذي يحوله إلى كود تنفيذي .

٢- بدايتك مع Microsoft Visual C++ 6.0 :

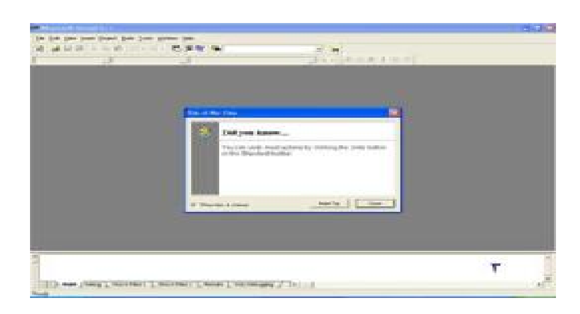
عند تحميل Microsoft visual studio 6.0 يكون من ضمن محتوياتها Microsoft Visual C++ 6.0 و Microsoft Visual basic 6.0 وغيرها .مايهمنا هنا حقاً هي Microsoft Visual C++ 6.0 فعند تشغيله لأول مرة من خلال رمزه الموضح بالشكل ١ :



عند الضغط عليه سوف تظهر لنا هذه النافذة التي ماتلبث ثواني وإلا وهي قد أختفت كما في الشكل ٢:



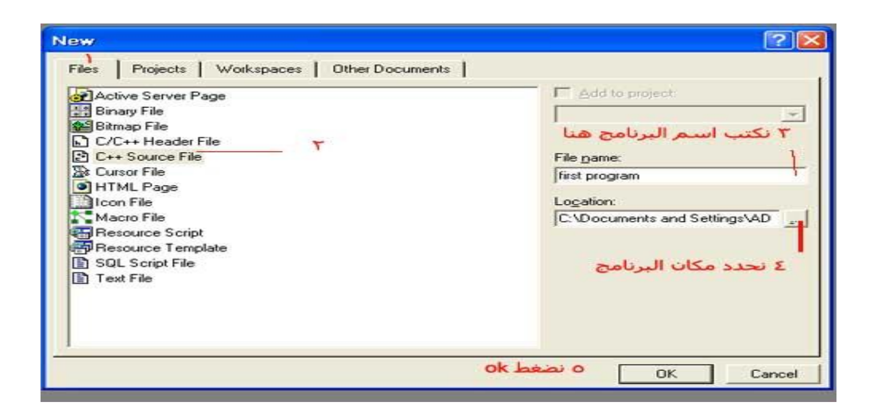
بعد ذلك ستظهر لنا نافذة البرنامج الرئيسية وستظهر نافذة فرعية صغيرة نقوم بالضغط على close لإغلاقها كما في الشكل ٣ :



حتى نبدأ طريقنا للدخول لعالم كتابة الكود البرمجي لابد من إتباع الخطوات التالي :

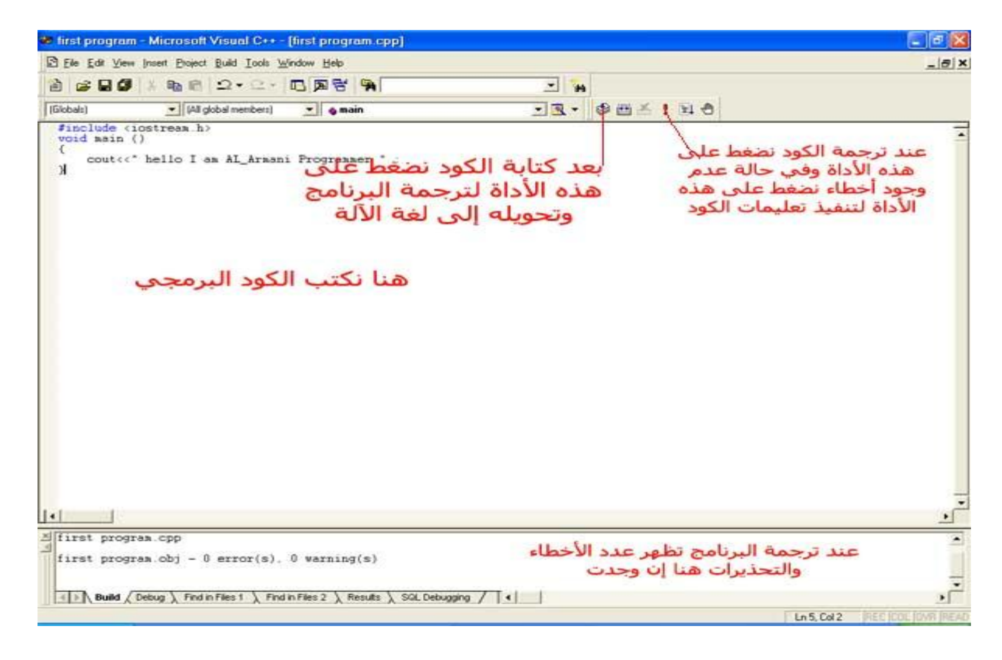
١- من قائمة file نختار new
٢- من قائمة new التي ستظهر نختار files ثم نحدد c++ source file
٣- نقوم بكتابة أسم الملف ونحدد مكانة ثم نضغط موافق .

نلاحظ كما في الشكل :

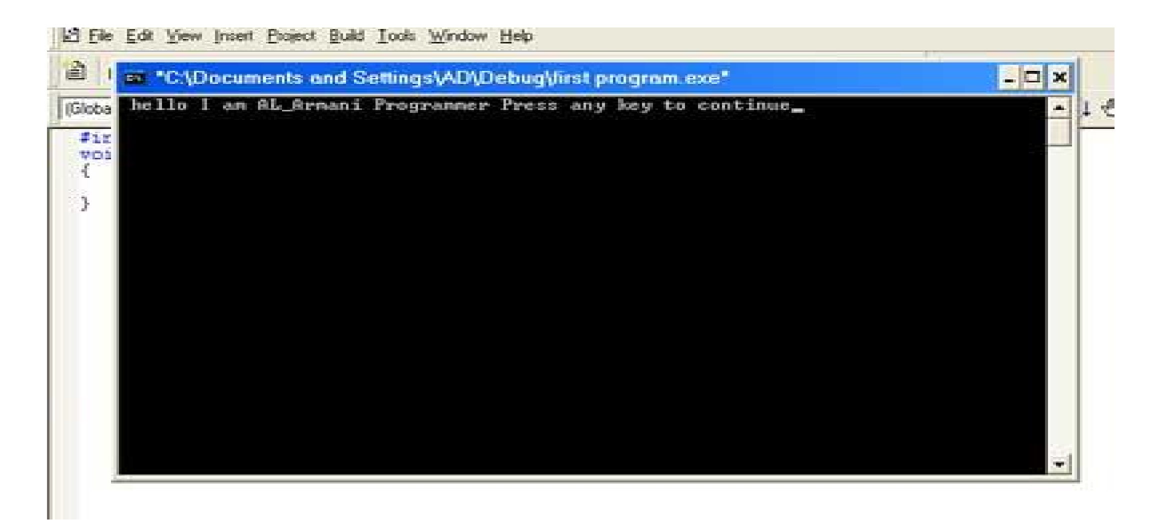


كيفية كتابة الكود :

هنا للتوضيح قمنا بكتابة كود عبارة عن طباعة رسالة ترحيب .. والشكل يوضح كيفية كتابة البرنامج والترجمة والتنفيذ ..



بعد تنفيذ البرنامج سوف تظهر لنا شاشة التنفيذ وهي الشاشة السوداء كما في الشكل التالي :



ملاحظة :

عند الضغط على رمز الترجمة compile ورمز التنفيذ execute وظهرت لنا نوافذ صغيرة بها نعم أم لا نختار نعم .

٣- كتابة البرنامج بلغة السي ++ :

قبل كتابة البرنامج يجب ان ندرك بأن لغة السي++ حساسة جداً لحالة الحروف الصغير والكبيرة ، فكلمة MOH تختلف جدا عن كلمة mOh و MOh . ويجب أن ندرك أيضا بأن كل الكلمات المحجوزة للغة جميع أحرفها صغيرة مثل void ، main ، include وغيرها . وعند ظهور أخطاء نستطيع معرفتها في النافذة الصغيرة السفلية والتي دائما تكون ظاهرة ، والاخطاء دائماً تكون إما نسيان كتابة الفاصلة المنقوطة أو كتابة إحدى الكلمات المحجوزة بحرف كبير .

البرنامج الأول : برنامج بسيط :

```
//Hello in my first program in c++ language

#include<iostream.h>

void main()

{

cout<<" I am AL_Armani Programmer ";

}
```

شرح البرنامج :

//Hello in my first program in c++ language

أي سطر في لغة السي++ يبدأ بـ // فهذا يعني بأنه عبارة عن تعليق أو ملاحظة يستخدم للتوضيح فالسي++ عندما يرى هذه العلامة يعرف بأنها ملاحظة فيقوم بتجاهلها .

أيضا إذا كانت الملاحظة مكونة من عدة سطور فيصعب أن نبدأ كل سطر بـ // لذا هذا نوع آخر من الملاحظات يسمى long comment وهي تبدأ بـ */ وتنتهي بـ /* ، مثلا :

```
/* Here we write every thins

As

Comments

*/
```

المبرمج العرماني

#include<iostream.h>

include # هو عبارة عن موجه للمترجم لكي يقوم بإدراج ملف رأسي ضمن الملف الأساسي . ويجب أن يوضع الملف الرأسي بين هاتين العلامتين <>

iostream.h (حرف الـ i يدل على الإدخال input وحرف الـ o يدل على الإخراج output)

هو عبارة عن ملف يحتوي على دوال وتصريحات نحتاج لإستخدامها في البرنامج وهذا الملف الرأسي ضروري كتابته إذا أردنا إستخدام cout وcin وهي أهداف للإخراج والإدخال .

void main()

هذا السطر هو عبارة عن الدالة الأساسية في أي برنامج مكتوب بلغة السي++ وأول ماينفذ من البرنامج هو هذه الدالة فهي ضرورية وتحتوي على كل شي من نداءات لدوال أخرى وتصريحات وأهداف وغيره .

وكلمة void تعني ان هذه الدالة لاترجع أي قيمة . والأقواس الدائرية مهمة جدا فهي تعني أن هذا السطر هو عبارة عن دالة وليست متغير .

{

}

هذا القوسان هما جسم الدالة الرئيسية body of function ونكتب بداخله الكود المراد ولايصح كتابة أي كود خارج هذين القوسين وإلا سيظهر خطأ error .

المبرمج العرماني

cout<<" I am AL_Armani Programmer ";

cout هو عبارة عن هدف يقوم بعرض رسالة معينة أو إظهار قيمة متغير معين . أما الإشارة << فهي تقوم بتوجيه مايكتب على يمينها إلى الهدف الذي على يسارها .

" I am AL_Armani Programmer "; هذه هي الرسالة المراد ظهورها على شاشة التنفيذ ويجب ان تكتب بين علامتي التنصيص المزدوجة . أيضا أي جملة في لغة السي++ لابد وأن تنتهي بعلامة شبه فاصلة semicolon

وهذه الجملة تسمى ثابتا حرفيا .

المبرمج العرماني

البرنامج الثاني : جمع عددين :

```
/* hello everyone

This program to add two integer number

*/

#include<iostream.h>

void main ()

{

int number1;

int number2;

int number3;

cout<<" please enter your first number \n";
```

```
cin>>number1;
cout<<" please enter your second number \n";
cin>>number2;
number3=number1+number2;
cout<<" the sum of number1 and number2 which you entered is : \n" ;
cout<<number3;
}
```

شرح البرنامج :

نلاحظ بأننا أستخدما علامة الملاحظة الطويلة لأنها تتكون من عدة سطور .

int number1;

هذا السطر هو عبارة عن تعريف لمتغير من نوع int أي عدد صحيح واسمه number1 ونلاحظ ايضا بان الجملة تنتهي بشبه فاصلة كما هو المعتاد لنهاية أي جملة في السي++ . عرفنا هذا المتغير ليمثل الرقم الأول المراد جمعه مع الرقم الثاني .

int number2;

عرفنا هذا المتغير ليمثل الرقم الثاني المراد جمعه مع الرقم الأول .

int number3;

عرفنا هذا المتغير ليمثل حاصل جمع الرقم الاول والثاني .

إذن الصيغة الرئيسية لتعريف المتغيرات هي :

Datatype variablename ;

المبرمج العرماني

وقفة مع المتغيرات :

المتغير هو جزء مهم جدا في الكود البرمجي وهو عبارة عن اسم يقوم بأخذ قيمة معينة سواءا عددية أو حرفية وهذه المتغيرات تخزن في الذاكرة فعندما نعرف متغير ما فأنه يحجز له مكان في الذاكرة يحمل اسمه .

أنواع المتغيرات type of variables :

١- Integer variable
عبارة عن متغيرات عددية صحيحة سالبة وموجبة حجمه في الذاكرة 2byte
٢- Long variable
عبارة عن متغيرات عددية صحيحة ولكن حجمها في الذاكرة 4byte
٣- Character variable
عبارة عن متغيرات حرفية حجمه في الذاكرة 1byte
٤- Float variable

عبارة عن متغيرات حقيقية عشرية ذات الفاصلة العائمة وحجمه في الذاكرة 4byte ، وهناك أيضا أنواع للمتغيرات الحقيقية العشرية والتي هي أكبر حجماً من الـ float وهي double ، longdouble

الشروط التي يجب مراعاتها عند تسمية المتغير :

١- يجب أن لا يبدأ أسم المتفير بفراغ أو برقم ويمكن أن يبدأ بشرطة سفلى _ .
٢- يمكن إستخدام الأحرف الكبيرة والصغير معاً .
٣- لايمكن إستخدام الكلمات المحجوزة في لغى السي++ مثل void ، int وغيرها .
٤- يفضل أن يعبر إسم المتغير عن محتواه .

المبرمج العرماني

```
cout<<" please enter your first number \n";

cout<<" please enter your second number \n";
```

هذان السطران هما عبارة عن رسالة للمستخدم ليقوم بإدخال الرقمين الأول والثاني المراد جمعهما .

والعلامة n\ تعني النزول لسطر جديد بعد هذه الجملة وهي تضيف تنظيم لشكل البرنامج حت لا يحصل تزاحم .

المبرمج العرماني

```
cin>>number1;

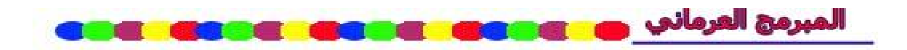
cin>>number2;
```

هذان السطران هما لعملية إدخال الرقم الأول والثاني وتخزين الرقم الأول في المتغير الذي عرفناه number1 والثاني في number2

المبرمج العرماني

```
number3=number1+number2;
```

هذا هو عبارة عن متغير اسمه number3 وقيمته عبارة عن تعبير expression حاصل جمع الرقمين .




```
cout<<" the sum of number1 and number2 which you entered is : \n" ;
```

هذه عبارة عن رسالة تظهر قبل عرض النتيجة .

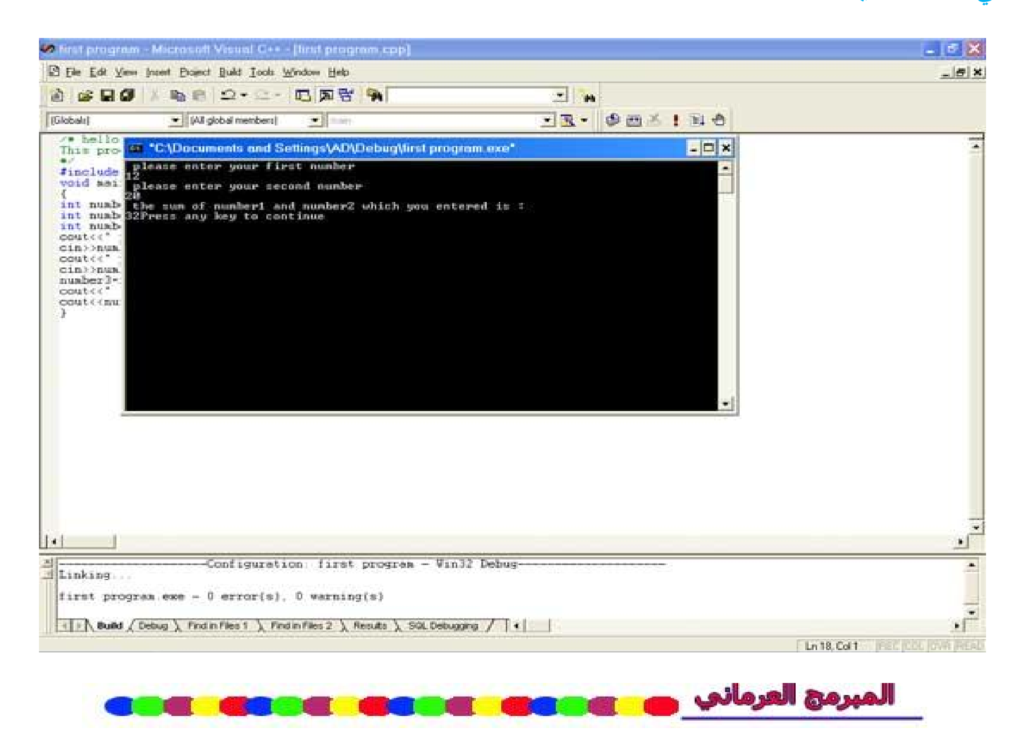
المبرمج العرماني

```
cout<<number3;
```

هنا يقوم بعرض قيمة number3 والذي هو حاصل جمع الرقم الأول والثاني .

وطريقة تنفيذ البرنامج أي عند ظهور الشاشة السوداء سوف تظهر رسالة تطلب من المستخدم إدخال الرقم الأول وبعد إدخاله للرقم يضغط على مفتاح enter لوحة المفاتيح ، بعد ذلك ستظهر رسالة تطلب منه إدخال الرقم الثاني وبعد إدخاله يضغط على مفتاح enter لتظهر له نتيجة جمع الرقمين .

وهذه هي شاشة التنفيذ :



المبرمج العرماني

أخيراً يمكننا أن نضع قيم ثابتة للرقمين الأول والثاني بدون إدخال من المستخدم عن طريق كتابة :

```
int number1=23;
int number2=10;
```



البرنامج الثالث : حساب مساحة الدائرة :

```
//program which calculate the area of circuit
#include<iostream.h>
Void main()
{
Float rad;
Const float PI=3.14;
Cout<<"enter radius of circuit :";
Cin>>rad;
Float area =PI*rad*rad;
Cout<<" The area of circuit is :"<<area;
```

```
}
```

شرح البرنامج :

Float rad;

هنا قمنا بتعريف متغير من نوع float وقمنا بتسميته rad ليمثل نصف القطر .

المبرمج العرماني

Const float PI=3.14;

كلمة const تعني ثابت فهنا قمنا بتعريف متغير ثابت من النوع float وإعطائه قيمة ثابتة و هي قيمة ثابتة رياضياً .

المبرمج العرماني

cout<<"enter radius of circuit :";

هنا لعرض رسالة توضيحية للمستخدم ليقوم بإدخال نصف القطر .

المبرمج العرماني

Cin>>rad;

هنا يقوم بإدخال نصف القطر .

المبرمج العرماني

Float area =PI*rad*rad;

هنا قمنا بتعريف متغير من النوع float واسمه area و هو يمثل المساحة وقيمة هذا المتغير هو عبارة عن ناتج المعادلة

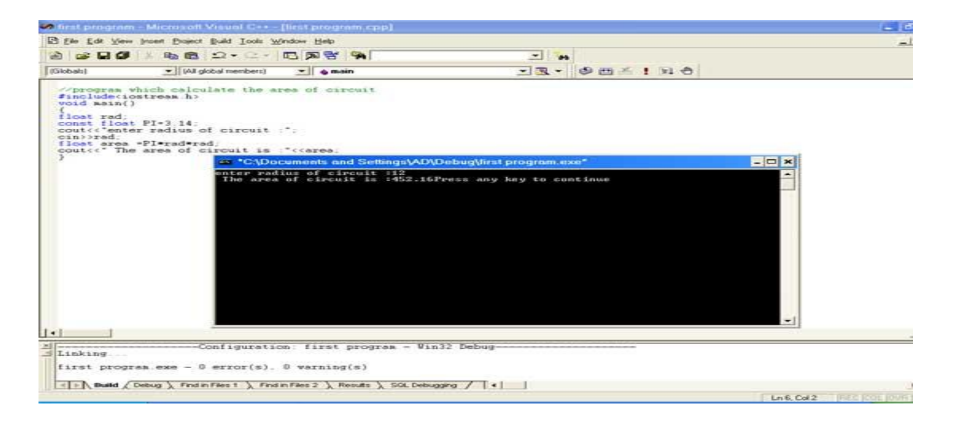
PI*rad*rad

المبرمج العرماني

Cout<<" The area of circuit is :"<<area;

هنا يعرض رسالة توضيحية قبل إظهار المساحة بعد حسابها .

وهذه هي شاشة التنفيذ :



البرنامج الرابع : إستخدام الحلقة لحساب حاصل جمع الأعداد من ٠ إلى ٤ :

الهدف من إستخدام الحلقة loop هي تكرار تنفيذ جزء من البرنامج عدد من المرات فهنا لابد أن يكون لدينا شرط معين إذا تحقق يتم تنفيذ التكرار وإلا سيتوقف .

```
#include<iostream.h>

void main ()

{

int sum=0;

for(int i=0; i<4; i++)

{

sum+=i;

}

cout<<" the sum of number from 0 to 4 is : \n " << sum ;

}
```

شرح البرنامج :

```
int sum=0;
```

قمنا بتعريف متغير من النوع int وأعطيناه القيمة صفر لأن متغير لحفظ مجموع الأعداد من ٠ إلى ٤ ويجب علينا تصفيره في البداية .

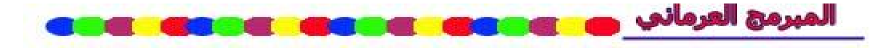
```
for(int i=1; i<=4; i++)
```

هذه هي حلقة التكرار ويعد الأمر for من أوامر الحلقات والأكثر إستخداماً ونستخدمها عندما نعرف عدد تكرار تنفيذ الجزء ، والضيغة العامة لها كما يلي :

for(القيمية الإبتدائية للعداد ; الشرط ; زيادة العداد)

نستطيع تعريف المتغير العداد وهو i داخل الأقواس الدائرية للحلقة أو ماقبلها أي بجانب تعريف الـ sum

هنا قمنا بتعريف i كعداد وأعطيناه قيمة ابتدائية واحد وبعد ذلك وضعنا شبه الفاصلة ثم الشرط للتكرار وهنا نريد أن يكرر أربعة أعداد من الواحد إلى الأربعة وبعد ذلك يقوم بالزيادة .



{

```
sum+=i;
}
```

هنا الأقواس تمثل جسم الحلقة أي كلما كان الشرط صحيح نفذ مابداخل هذا الجسم .

فهنا كلما كان الشرط صحيح كلما جمع قيمة العداد مع قيمة sum .

المبرمج العرماني

```
cout<<" the sum of number from 0 to 4 is : \n " << sum ;
```

هنا قمنا بإظهار النتيجة النهائية لحاصل جمع الأرقام ، وعمدنا إلى كتابة هذا السطر خارج جسم الحلقة لكي يقوم بعرض النتيجة النهائية بشكل صحيح أي بعد إنتهاء الحلقة .

المبرمج العرماني

طريقـــة سيـــر الحلقــة :

عند الدخول إلى الحلقة للمرة الأولى تكون قيمة العداد ١ ويرى هل ١ أقل من أو يساوي ٤ ، نعم يقوم بالدخول إلى جسم الحلقة ويقوم بجمع ١ و هي قيمة العداد على الصفر و هي قيمة الـ sum ويصبح الـ sum بواحد ، بعد ذلك يرجع إلى جملة الـ for ويقوم بزيادة العداد بواحد ليصبح قيمته ٢ ثم يرى هل الشرط تحقق أم لا ، نعم يدخل إلى جسم الحلقة ويقوم بجمع ٢ و هي قيمة العداد على ١ و هي قيمة الـ sum ليصبح الـ sum قيمته ٣

و هكذا إلى أن تصبح قيمة العداد أكبر من ٤ أي ٥ فيخرج من جسم الحلقة ليجد جملة طباعة الـ sum .

المبرمج العرماني

البرنامج الخامس : إستخدام حلقة التكرار while :

يستخدم الأمر while إذا كنا لا نعلم عدد مرات التكرار .

هذا البرنامج يقوم بإدخال أي رقم ماعدا الصفر بحيث إذا قام المستخدم بإدخال الصفر يتم الخروج من البرنامج .

```
#include<iostream.h>
void main ()
{
int r ;
Cout<<" you can enter any value to exit enter 0 ";
while ( r!=0)
{
cout<<"\n";
Cin>>r;
```

```
}
}
```

شرح البرنامج :

int r ;

قمنا بتعريف متغير اسمه r من النوع الصحيح وقيمة هذا المتغير يقوم بإدخالها المستخدم .

Cout<<" you can enter any value to exit enter 0 ";

هذه الرسالة توضيحية للمستخدم ليقوم بإدخال أي عدد ماعدا الصفر.

while (r!=0)

هنا طريقة كتابة أمر التكرار while وبداخل الأقواس الدائرية كتبنا الشرط بأن القيمة المدخلة لاتكون صفراً .

```
{
cout<<"\n";
Cin>>r;
}
```

هذا يعتبر جسم الحلقة وبداخلها جملة ;cout <<"\n" وهي للنزول إلى سطر جديد بين الأرقام المدخلة .

Cin>>r;

هذه الجملة لإدخال قيمة المتغير وهي أي عدد ماعدا الصفر حسب شرط الحلقة لكي تقوم بالتكرار .

البرنامج السادس : إستخدام حلقة التكرار do-while :

هذه الحلقة تختلف عن حلقة التكرار while في أنها تنفذ الجملة ثم تنظر إلى الشرط .

البرنامج هذا يقوم بضرب عددين وشرط التوقف هو إدخال حرف a .

```
#include<iostream.h>
void main()
{
long d1 , d2;
char let;
do{
cout<<"enter the first number\n";
cin>>d1;
cout<<"enter the second number\n";
cin>>d2;
cout<<" the result of multiple two numbers is :\n";
cout<<d1*d2;
cout<<"\n if you want to do again press any letter or number unless a \n";
cin>>let;
cout<<"\n";
}while(let!='a');
}
```

شرح البرنامج :

long d1 , d2;

هنا قمنا بتعريف متغيرين ليمثلا الرقمين المراد ضربهما وهما من النوع long لأن دائماً حاصل ضرب الأرقام يكون عدد كبير وبإمكانك تعريفهم على انهم int ولكن الأفضل في الضرب هو long ، أيضاً قمنا بتعريف المتغيرين للإختصار بسطر واحد عن طريق الفاصلة بدلاً من كتابة سطرين كالتالي :

Long d1;

Long d2;

char let;

هنا عرفنا متغير من النوع الحرفي او الرمزي ليمثل قيمة الشرط المدخلة وكما نعلم بأن النوع char يقبل الأرقام والحروف ولكن لانستخدمه للأرقام عندما نريد إجراء عملية حسابية عليهم .

المبرمج العرماني

do{

من هنا تبدأ حلقة التكرار do-while وصيغتها كالتالي :

Do{

هنا نكتب الجمل المراد تنفيذها أو الأوامر ..

}while(condition); هنا بعد قفل قوس الحلقة نكتب الـwhile ثم قوسين دائريان وبداخلهما الشرط ولاننسى بالنهاية ان نضع فاصلة منقوطة .

المبرمج العرماني

```
cout<<"enter the first number\n";

cin>>d1;

cout<<"enter the second number\n";

cin>>d2;

cout<<" the result of multiple two numbers is :\n";

cout<<d1*d2;
```

الجمل الستة العليا هي عبارة عن :

١- رسالة توضيحية للمستخدم ليقوم بإدخال الرقم الأول
٢- إدخال الرقم الأول
٣- رسالة توضيحية للمستخدم ليقوم بإدخال الرقم الثاني
٤- إدخال الرقم الثاني
٥- رسالة توضيحية للمستخدم قبل عرض نتيجة حاصل الضرب
٦- إظهار أو عرض حاصل ضرب الرقم الأول في الرقم الثاني

المبرمج العرماني

```
cout<<"\n if you want to do again press any letter or number unless a \n";
cin>>let;
cout<<"\n";
```

هذه الجمل عبارة عن :

١- رسالة توضيحية للمستخدم ليقوم بإدخال أي رقم أو أي حرف لكي يعيد تنفيذ الأوامر من جديد أي إعادة عملية إدخال الرقمين وضربهما بإستثناء حرف الـ a فإذا قام بإدخاله فإنه يخرج من الحلقة .
٢- إدخال الحرف أو الرقم وتخزينه بالمتغير الذي عرفناه مسبقاً وأعطيناه الاسم let
٣- الجملة الأخيرة هي لتنظيم البرنامج للنزول لسطر جديد .

المبرمج العرماني

```
}while(let!='a');
```

هنا نلاحظ قفل قوس جسم الحلقة ومن ثم الشرط وهو أن الحرف الذي تم إخاله لتكرار الحلقة لايكون حرف a ونلاحظ بأن حرف تم وضعه بين علامة تنصيص مفردة ، وفي نهاية الشرط تم وضع فاصلة منقوطة .

المبرمج العرماني

البرنـامج السابع : إستخدام أمر الشرط if :

في بعض البرامج قد نحتاج إلى إتخاذ قرارات معينة والتي من خلالها تحدد سير البرامج وهذا يتم من خلا أمر الشرط والذي يكون محقق أو غير محقق لإكمال مسار جزء معين من البرنامج أو البرنامج ككل .

هذا البرنامج يقوم بإظهار جميع الأرقام المدخلة من قبل المستخدم والتي لابد أن تكون أصغر من ٥٠ .

```
#include<iostream.h>
Void main ()
{
Int k;
Cout <<" enter your number : \n ";
Cin>> k;
If ( k<50)
Cout<<" the number is less than 50 ";
}
```

شرح البرنامج :

كل الجمل قد مرت علينا أثناء شرح البرامج السابقة ماعدا :

if (k<50)

وهذه الجملة هي شرط أن تكون الأرقام أقل من ٥٠ لكي يكمل البرنامج .

المبرمج العرماني

البرنامج الثامن : إستخدام عبارة switch :

نستخدم هذه العبارة عندما تكون لدينا خيارات متعددة .

هذا البرنامج عبارة عن آلة حاسبة للجمع والضرب والطرح والقسمة .

```
#include<iostream.h>

Void main ()

{

Float num1,num2;

Char symbol;

Float result;

Cout<<"please enter your two numbers :\n";

Cin>>num1;

Cout<<"\n";

Cin>>num2;

Cout<<"please enter the symbol of operator which you want to execute:\n":

Cin>>symbol;

Switch(symbol)

{

Case '+':

Result=num1+num2;

Break;

Case '-':

Result=num1-num2;

Break;
```

```
Case '*':
Result=num1*num2;
Break;
Case '/':
If(num2!=0.0)
Result=num1/num2;
Else
{
Cout<<"error happned , you devided by zero ";
Result=0.0;
}
Break;
Default:
{
Cout<<"you entered unknown symbol ";
Result=0.0;
}
Cout<<"the result is : \n" <<result;
}
}
```

شرح البرنامج :

```
Float num1,num2;
Char symbol;
Float result;
```

١- قمنا بتعريف متغيرين من النوع float لأن عملية القسمة ربما تكون بباقي ويكون الناتج أعداد عشرية .
٢- قمنا بتعريف متغير ليمثل الإشارة المدخلة سواءا ضرب طرح قسمة جمع .



٣- أخيراً عرفنا متغير ليقوم لتخزين نتيجة العملية الحسابية .


```
Cout<<"please enter your two numbers :\n";
Cin>>num1;
Cout<<"\n";
Cin>>num2;
Cout<<"please enter the symbol of operator which you want to execute:\n":
Cin>>symbol;
```

أعتقد أخي القارئ بأن هذه الجمل أصبحت معروفة لديك ، فهذه مجرد رسائل توضيحية للمستخدم ليقوم بالإدخال وأيضا إدخال الرقمين والإشارة الحسابية .


```
Switch(symbol)
{
```

من هنا يبدأ إستخدامنا لعبارة switch بحيث نعطيها الإشارة المدخلة وهي تقوم بتوجيهها إلى حالتها .


```
Case '+':
Result=num1+num2;
Break;
```

هذه تمثل الحالة الأولى وهي إذا كانت الإشارة المدخلة جمع فإنه يتم الدخول لهذه الحالة نلاحظ أولا كتبنا كلمة case وهي تعني حالة ثم عنوان الحالة بعدها وهنا نقصد علامة الجمع وقمنا بوضعها بين علامة تنصيص مفردة ثم نقطتين عموديتين .

وقمنا بكتابة معادلة الجمع وتخزين النتيجة في المتغير result .

أخيراً كتبنا كلمة break وهي تعني الخروج من الـ switch نهائياً .

```
Case '-':
Result=num1-num2;
Break;
Case '*':
Result=num1*num2;
Break;
```

هذه هي حالتي الطرح والضرب وهي بنفس طريقة حالة الجمع .

المبرمج العرماني

```
Case '/':
If(num2!=0.0)
Result=num1/num2;
Else
{
Cout<<"error happned , you divided by zero ";
Result=0.0;
}
Break;
```

هذه هي حالة القسمة وهي نفس الحالات السابقة ولكن قمنا بإضافة شرط لها وهو ان المقسوم عليه يجب ان لايكون صفرا (If(num2!=0.0

وقمنا بكتابة الصفر عشريا لاننا عرفنا الارقا في الاعلى على انها float .

فإذا تحقق الشرط وكان المقسوم عليه ليس صفرا فإن عملية القسمة سوف تتم .

```
Result=num1/num2;
```

وإذا لم يتحقق الشرط فإنه يقوم بالدخول إلى else

وتظهر رسالة بأنه تم القسمة على صفر .

ويوضع ال result بصفر .

ويتم الخروج من switch .

```
Cout<<"the result is : \n" <<result;

}
```

هنا يتم طباعة النتيجة .

أخيراً : يمكننا أن نكرر العمليات الحسابية بإضافة حلقة تكرارية والأفضل أن نستخدم do-while قبل عملية إدخال الرقمين والإشارة ، كما في البرنامج الذي تم شرحه مسبقاً بأنه لو تم إدخال حرف معين فإنه يتم الخروج من الحلقة .

٤- مقدمة عامـــة عن البرمجة موجهة الهدف (Introduction to Object Oriented Programming)

تعتبر كتابة البرامج الطويلة والضخمة أمر معقد ، ولكن نستطيع حل هذا التعقيد بواسطة تقسيم هذا البرنامج إلى وحدات أو أجزاء functions وكل جزء يقوم بعملية معينة ، ولربما ضربنا مثلاً في حياتنا : وهو أننا لا نستطيع أكل بطيخة دفعة وحداة وإلا سيكون الأمر متعب للغاية للأسنان والفم وللمعدة بالطبع ولكن يمكننا تقطيعها إلى قطع صغيرة مما يسهل علينا عملية الأكل (بالعافية على الجميع الظاهر فتحت شهيتكم) ، وهذا ينطبق على كتابة البرامج الكبيرة فكلما قمنا بتجزئته إلى وحدات كلما كان أصبحت عملية الكتابة سهلة جداً وأيضا حتى عندما نواجه مشكلة تساعد عملية التجزئة إلى حل المشكلة من خلال تحديد مكانها بأي جزء هي .

عند التحدث عن لغة C++ لابد لنا جبراً أن نتحدث ولو بإختصار عن مفهوم البرمجة موجهة الهدف .

ماهي البرمجة موجهة الهدف (Object Oriented Programming) :

هي أسلوب برمجي يعتمد على توحيد أو جمع البيانات data والدوال functions في وحدة برمجية تسمى الهدف object ، فلكي نستطيع الوصول إلى البيانات لابد لنا من إستخدام الدوال والتي تسمى دوال العضوية member functions . والبرنامج المكتوب بلغة السي بلس بلس يمكن أن يحتوي على العديد من الأهداف ، والأهداف هي عبارة بالتالي عن أعضاء من الصنف class ، إذن نستطيع القول بأن الصنف هو عبارة عن مجموعة من الاهداف ، فمثلا : الحشرات ، الزواحف ، الثدييات هم أعضاء في صنف الحيوانات .

أخيراً إن أسلوب البرمجة موجهة الهدف يسمح لنا بتنظيم البرنامج لكي يسهل من فهمه ، أيضاً يسمح لنا بتعريف أكثر من نوع للبيانات .

خصائص اللغات التي تتميز بالبرمجة موجهة الهدف :

١- الوراثة Inheritance

سوف نقوم بشرح هذه الخاصية من خلال ضربنا لهذا المثل : لدينا صنف إسمه الحيوانات وهو يمثل الخصائص العامة لجميع الحيوانات نستطيع من خلاله أن نشتق صنف جزئي يرث منه صفات عامة يسمى الطيور والتي لها خصائص خاصة بها من الجناحان ، المنقار والتي لاتوجد في الزواحف والثدييات . إذن فالصنف المسمى بالحيوانات يطلق عليه الصنف الرئيسي base class والصنف الجزئي المسمى بالطيور يطلق عليه الصنف المشتق derived class.

٢- إعادة الإستخدام Reusability

بعد كتابة البرنامج يستطيع المبرمجون إضافة مميزات له وإمكانيات بواسطة الإشتقاق لصنف جدي وما إلى ذلك وهذا مايسمى بإعادة الإستخدام .

٣- تغليف البيانات Encapsulation

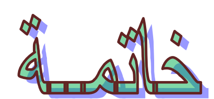
وهي عملية إخفاء البيانات لمنع الوصول إليها من دوال غير مسمح لها .

بقي أن نقول فيما يخص البرمجة موجهة الهدف بأن عندما نواجه مشكلة ما فإننا نسأل كيف سنقسمها إلى أهداف وليس إلى توابع .

الفرق بين لغة C و C++ :

تعتبر لغة السي لغة إجرائية أي أنها عبارة عن سلسلة من الأوامر التي ينفذها الكومبيوتر فهي ليست من اللغات التي تعتمد على البرمجة موجهة الهدف .

(السي ، باسكال ، فورتران ، بيسك هي لغات لاتعتمد على البرمجة موجهة الهدف) .



أخيراً أسأل الله العلي القدير أن يجعل هذا الكتاب لبنة من لبنات المكتبة العربية العلمية وأن ينفع به العرب والمسلمين ولو بشكل بسيط ... وأتمنى من الله أن يمدني بالصحة والعافية لإكمال مسيرة التأليف للجزء الثاني من هذا الكتاب وعند وجود أي ملاحظات حول الكتاب يمكنكم مراسلتي عبر الإيميل .

والله من وراء القصد .

أخوكم : المبرمج العرماني

جمادي الأول لعام ١٤٢٨ هـ

